

(R)

Some signs :

"=" "<-" assign a value to an object in R

ex. $x = 11$
 $x <- 11$

print (x) or
type the name x to see what is stored

ls () to know everything that stored in the workspace memory

rm (y) to remove an object

Note : number can not appear as the first character in the object's name
ex. $1x <- 22 \rightarrow$ error

xx = "main" | assign character values to objects

yy = "1" \Rightarrow 1 is a character, not number.

Note : we can perform arithmetic operations (* + - / ^ sqrt(y) log(y) exp(y) log2(y) absolute number abs(-14))

z <- x+y | store in z

Note : if we enter incomplete command, R will follow that up with (+) plus sign. Once we add it, return the answer

press the arrow up key \rightarrow brings up the previous command
down \rightarrow move forward

can make notes with number signs or #

ignore everything that follows after number signs or

concatenate
c(...)
 $x1 <- c(1, 3, 5, 7, 9)$ create a vector or column of numbers or characters.

2:7 Create a sequence in integer values

seq(from=1, to=7, by=1)

rep(1, times=10) create a vector of repeated numbers or characters

Note : can be combined ex. $rep(seq(from=2, to=5, by=0.5), times=4)$

we may add/subtract value to each element of the vector.

two vectors with the same length, we may add/subtract/*div corresponding elements

y[3] the third element

y[-3] except the third element

y[c(1, 5)] extract the first & third element

y[-c(1, 5)] extract except the 1st & 3rd element

matrix () Create a matrix of values

matrix (c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow=3, byrow=TRUE)

byrow=FALSE

elements

will be entered in a row-wise fashion

$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

will be entered in a column-wise fashion

$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$

Note : [] can also be used to extract multiple elements from the matrix.

ex. $mat[1, 2] \rightarrow$ extract the element in the first row and the second column.

$mat[2,] \rightarrow$ extract all elements in row 2

we can perform element-wise + - /*

ex. $mat * 10$

强制把 x 变成 numeric 的性质

as.numeric(x)

"2L" 整数 (integer)

(I)

vector 向量 Component 分量
element 元素

matrix 矩阵

data frame 资料框架

list 列表 Item 项

character 字符

string 字符串

Character vector 字符串向量

argument 引数

parameter 参数

save data files as comma separated value (.csv) or tab delimited text (.txt)

read.csv (file.choose(), header = TRUE) first row of datasets are variable names or headers.
menu popping up, allow us to select the file directly)

} read csv files

read.table (file.choose(), header = TRUE, sep = ",")
how the data values are separated.

read.delim (file.choose(), header = TRUE)

} read txt files

read.table (file.choose(), header = T, sep = "\t")
to know this is a tab-delimited file

write.table (DataToExport, file = "Exported File Name.csv", sep = ",")
name of the new file
data to be exported from R (blank) to specify the format "t"

} export dif. types of files

write.table (DataToExport, file = "Exported File Name.csv", row.names = F, sep = ",")
could add the path to save in dif. working directory get rid of the row names

notes: export file w/ the same name will overwrite the previously saved file w/o giving us warning.

write.csv (DataToExport, file = "/Users/Erika/Downloads/Exported File Name.csv", row.names = F)
doesn't need sep = ","

} export to csv file.

write.csv2 will use comma for decimal points and semi-colon for separator

dim (LungCapData) dimension of the data (number of rows and columns)

head () view portion of data (first 6 rows)

tail () to see the last 6 rows

names () to check the variable names for the data

mean (LungCapData\$Age) to extract the variable in the data

attach (LungCapData) $\left\{ \begin{array}{l} \oplus \text{ able to call in variables by their names, no need to extract by \$} \\ \ominus \text{ put in R's workspace memory \& can be overwritten more easily, hang around until we erase them from R's memory.} \end{array} \right.$

detach (LungCapData) unattach the data (erase the memory)

class (Height) check the type of a variable ex. numeric, integer, factor/categorical

levels (Smoke) [for factors' variable] to ask what different levels for this factor \Rightarrow "Yes" "No"

summary (LungCapData)

provide a summary it believes is appropriate for each of the variables.

as.factor (x)

convert a data to a categorical variable or a factor (summary: will check frequencies, instead of mean, median...)

length (Age) to ask abt the number of observations in a vector or a variable

mean (Age [Gender == "female"]) to represent the meaning of equality in a mathematical sense.

\rightarrow calculate the mean Age only for females.

MaleOver15 <- LungCapData [Gender == "male" & Age > 15,]

\rightarrow pull out a subset of data for males who are over 15 years old.

`temp <- Age > 15` create a logic vector or variable asking R whether or not age > 15.

answers to \Rightarrow TRUE/ FALSE

temp2 <- as.numeric(Age > 15) returning to 0 (for False) and 1 (for TRUE)

`cbind (LungCapData, FemSmoke)` attach vectors or matrices in a column-wise fashion

row-wise fashion

remove the lists of all objects in the workspace memory.

or Session \rightarrow Clear Workspace

or Session → Clear Workspace
 转格式: Tools → global options → code → saving → default text encoding: UTF-8

不同 data type 但都方正

	# Matrix	1 2 3
	1	...
	2	...
	3	...

增加列

	# dataframe	1 2 3
	1	...
	2	...
	3	...

行是变数

list (): 较有弹性, 可以塞不同分量的向量
e.g. list (cell, "J")

for (i) {
 ↓
 index
}

迴卷

find current working directory

set working directory

setwd("~/Desktop/Project1") (if you don't want to specify the path, used ~)

or from the Menu : Session → Set Working Directory → Choose Directory.

save_image ("First Project, Rdata") save current workspace image (file) [no specify path = ✓ save in current directory].

or from the Menu : Session \rightarrow Save Workspace As ...

quit Rstudio

or File → Quit RStudio

load ("First Project. Rdata") load previous workspace image

OR

`load (file.choose())` OR Session → Load Workspace

(should be saved w/ .R)

script → pick up where they left off on a project & progressively build and refine code and analysis, easily reproduce

↓

↓
a set of commands that usually includes commenting on what each piece of code is intended to do. Analysis run earlier

File → New File → R script OR File → Open File

"Ruh"

Submit the scripts

"ctrl + Enter"

Code → Comment / Uncomment Lines take entire section of codes and add comments / remove hash lines (#)

"Tab" key return you a list of suggestions of what you may be looking for (when typing)

"Tab" key

install. packages ("epiR") install.packages (Tools → Install Packages)

library (epiR)

can use all the commands and functionality in "epiR" package

www.r-project.org → CRAN → select location

remove. packages ("epiR") delete or remove packages

10/22

library(stringr)

str_split 可切割里面的 component. Ex \Rightarrow `kk = "abcdefghijkl"`
`pp = str_split(kk, "e")` pp is now "abcd" "fgh" "ijk"
if `hh = str_split(kk, c("d", "j"))` hh is now [1] "abc" "efghijk"
[2] "abcdeghi" "k"

paste0 combine the first & other component in the list.
Ex. `zz = paste0("Mary", yy[[1]][2])`
`zz = "Mary Wang"`

str_locate(text1, "...") locate the position of patterns in a string.

str_sub(..., ...) extract & replace substrings from a character vector.

file.rename("#from", "#to", ".pdf") rename a file

list.files(getwd(), full.names = F) produce a character vector of the names of files / directories in the named directory.

Apply functions less lines of code (less chance of coding error) & faster than a simple loop.

Apply(x, margin, fun, ...)

x: the object we would like to apply some function to

margin: specifies if function applied to rows or columns (1 = row, 2 = column)

fun: the function we like to apply

...: where we specify other arguments to send to the function

if there is NA:

Apply(X = StockData, MARGIN = 2, FUN = mean, na.rm = TRUE)
(max) remove missing values when applying the "mean" function to data.
(sum)

colMeans(StockData, na.rm = TRUE) calculate column means (more optimized than apply function)

apply(..., FUN = quantile, probs = c(0.2, 0.80), na.rm = TRUE) \rightarrow percentile

...., FUN = plot, type = "l")
to produce a line plot

_____, main = "stock", ylab = "Price", xlab = "Day") \rightarrow add some additional arguments

rowSums(StockData, na.rm = TRUE)

plot(apply(...), type = "l", ylab = "...", xlab = "...", main = "...") \rightarrow make a line plot
title

points(apply(...), pch = 16, col = "blue") add colours.

(by)

tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)
 \rightarrow let R know to simplify the results if possible.

X: variable or vector we will apply function to

INDEX: a grouping variable that is the same length as X and is used to create the subsets of data.

...: any additional arguments to be passed to the function.

or directly: `tapply(Age, smoke, mean)`

INDEX = list(Smoke, Gender): calculate mean Age based on Smoking status and Gender

mean(Age[Smoke == "yes" & Gender == "female"]) mean of age by subsetting smoke & gender

10/29

Sys.getlocale() 搜尋目前的 language

Sys.setlocale() setting 要 的 language

dim() dimension \rightarrow Column = ... Row = ...

any(aa, na.rm = T) check whether there is an element that is TRUE